REMARKS

The enclosed is responsive to the Examiner's Final Office Action mailed on March 30, 2005 and is being filed pursuant to a Request for Continued Examination (RCE) as provided under 37 CFR 1.114. At the time the Examiner mailed the Office Action claims 1-26 were pending. By way of the present response the Applicants have: 1) amended claim 23; 2) added no new claims; and 3) canceled no claims. As such, claims 1-26 are now pending. The Applicants would like the Examiner to note in the previous response there were typographical errors in the legends of the claims. The Applicants have appropriately labeled each legend as submitted herewith in this response. The Applicants respectfully request reconsideration of the present application and the allowance of all claims now presented.

Claim Rejections

35 U.S.C. 102(e) Rejections

The Office Action rejected claims 1-26 under 35 U.S.C. 102(e) as being anticipated by Kalafatis, et al., U.S. Patent 6,535,905 (hereinafter "Kalafatis").

Kalafatis describes "thread switching operation within a multithreaded processor" in certain situations. (See Kalafatis, Abstract.) For example, Kalafatis describes "that [when] the allocator [of a microprocessor] determines that insufficient resources ... are available for instructions (i.e., microinstructions) for a particular thread received from the queue ... the allocator 76 asserts a stall signal ...On the assertion of such a stall signal 82 for a particular thread, it may be desirable to perform a thread switching operation." (Kalafatis, Col. 13, lines 32-46.) Kalaftis is not describing relinquishing portions thread partionable resources but merely that if there are insufficient resources available for a particular thread it may be desirable switch to a different thread.

Kalafatis further describes "performing a thread switching operation within a multithreaded processor on the sequencing (or dispatch) of a branch instruction of a current thread (e.g., thread 0) from the instruction streaming buffer 106 to the instruction pre-decoder 108." (See Kalafatis, Col. 10, lines 25-39.) Kalaftis is not describing "to relinquish portions of said plurality of thread partionable resources" but merely thread switching on a branch instruction.

Kalafatis further describes "[i]n a multithreaded processor, where a page miss occurs for an instruction stream of a current thread, it may be advantageous to perform a thread switching operation so as to allow an alternative thread to utilize the latency introduced by the page walk operation. ... [A] determination is made as to whether a predetermined minimum quantity of instruction information (e.g., a predetermined minimum number of chunks) for an alternative thread (e.g., thread 1) are pending and available for dispatch from the logical partition 124 of the instruction streaming buffer 106." (Kalafatis, Col. 18, line 62 through Col. 19, line 53.) Kalaftis is not describing relinquishing portions thread partionable resources but determining if a minimum number of instruction information is available for an alternative thread to process.

Furthermore, while Kalafatis describes thread switching in certain situations, the discussion of partitioning is limited to the fact that the resources are partitioned in the multithreaded mode. See, e.g., Fig. 4 and discussion at Col. 7, ll. 6 – 61:

> The logical partitioning for two threads of the buffering (or storage) facilities of a functional unit may be achieved by allocating a first predetermined set of entries within a buffering resource to a first thread and allocating a second

> predetermined set of entries within the buffering resource to a second thread. Specifically, this may be achieved by providing two pairs of read and write pointers, a first pair of read and write pointers being associated with a first thread and a second pair of read and write pointers being associated with a second thread. The first set of read and write pointers may be limited to a first predetermined number of entries within a buffering resource, while the second set of read and write pointers may be limited to a second predetermined number of entries within the same buffering resource.

However, Kalafatis does not describe that thread switching should also cause the relinquishing of resources. The mere switching to another thread does not automatically cause the prior thread's partitioned resources to be released for use by the other thread. Indeed, if any thread switch could cause the relinquishing of resources, this could, for example, lead to very inefficient program execution because relinquishing and partitioning of processor resources may in some cases be a time consuming task. For example, thread switching may be done in the case where a thread is very active. See "The Forced Thread Change Logic (160)", Kalafatis, Col. 16, l. 51 – Col. 18, l. 40). To relinquish partitions of resources for an overactive thread may be inefficient and is not described by Kalafatis.

With respect to claim 1, Kalafatis does not describe "a plurality of thread partitionable resources that are each partitionable between a plurality of threads including a first thread and at least one other thread; logic to receive a program instruction from a first thread directing said processor to suspend execution of said first thread, and in response to said program instruction to cause the processor to suspend execution of the first thread and to relinquish portions of said plurality of thread partitionable resources associated with the first thread for use by other ones of said plurality of threads."

Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 1 requires. Claims 2-12 are dependent upon claim 1 and are allowable for at least the same reason.

With respect to claim 13, Kalafatis does not describe "receiving a first opcode in a first thread of execution; suspending said first thread for a selected amount of time in response to said first opcode; relinquishing a plurality of thread partitionable resources in response to said first opcode."

Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 13 requires. Claims 14-17 are dependent upon claim 13 and are allowable for at least the same reason.

With respect to claim 18, Kalafatis does not describe "a memory to store a plurality of program threads, including a first thread and a second thread, said first thread including a first instruction; a processor coupled to said memory, said processor including a plurality of thread partitionable resources and a plurality of shared resources, said processor to execute instructions from said memory, said processor, in response to execution of said first instruction to suspend said first thread and to relinquish portions of said plurality of thread partitionable resources."

Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 13 requires. Claims 14-17 are dependent upon claim 13 and are allowable for at least the same reason.

Unlike Applicants' other claims, claim 24 is a means plus function claim that must be interpreted under 35 USC 112, paragraph 6 to the structures and equivalents thereof discussed in the application. With respect to claim 24, Kalafatis does not describe "means for receiving a first instruction from a first thread; means for suspending said first thread in response to said first instruction; means for relinquishing a plurality of partitions of a plurality of resources; means for re-partitioning said plurality of resources after a selected amount of time." Kalaftis does not describe a means for relinquishing partitions or a means for re-partitioning these same resources.

Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 24 requires. Claims 25-26 are dependent upon claim 24 and are allowable for at least the same reason.

In light of the comments above, the Applicant respectfully requests the allowance of all claims.

CONCLUSION

For the reasons provided above, applicant respectfully submits that the current set of claims are allowable. If the Examiner believes an additional telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Daniel M. DeVos at (408) 720-8300.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 8/1/05

Daniel M. DeVos
Reg. No. 37,813

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300